Mike Gibson, Runjie Zhang
ECE 6332 – Calhoun
Design Project Proposal
15 Oct 2009

Design Project Proposal

Introduction

Field programmable gate arrays (FPGAs) are useful in a wide variety of applications. Their usefulness stems primarily from their flexibility and cost-efficiency. Compared to application specific integrated circuits (ASICs), FPGAs are typically easier to design and have lower production costs [2]. To facilitate increased configurability, however, FPGAs must consist of a large number of flexible circuit elements and as a result must consist of more transistors. This leads to increased overall power consumption when compared to similarly configured ASICs.

Leakage power consumption in this extra hardware accounts for a significant portion of the overall power consumption; as [2] suggests, this figure is as high as 35% of the total power consumption in 90nm technologies when available CLB utilization is 100%. This can limit the application space open to FPGAs, specifically prohibiting FPGA use in energy sensitive applications such as mobile applications [2]. Additionally, as FPGAs become smaller, this contribution will only increase. As outlined in [7], the portion of overall power consumption attributed to leakage power increases as transistor gate lengths decrease. In an effort to mitigate overall power consumption, recent research has yielded an FPGA that operates in the sub-threshold regime. Leakage power plays an even larger role in sub-threshold power consumption. As [1] indicates, leakage power begins to increase exponentially at low sub-threshold voltages limiting the power savings when operating below some optimum value of VDD. As FPGAs enter the sub-threshold regime, then, leakage power reduction will need to become an even more critical goal in order to realize maximum power savings.

Numerous works have investigated leakage reduction techniques for both general circuits and FPGAs in super-threshold operation. [3] and [4] developed models for estimating overall FPGA power consumption. [5] applied dual-Vt method to reduce leakage current in FPGAs and proposed algorithms to maximize the method's efficiency. [6] provided a number of generally applicable techniques for reducing sub-threshold leakage current. However, no previous work has focused on the application of these techniques to FPGAs operating in the sub-threshold regime.

Goals and Approach

Our project intends to determine the effectiveness of existing and novel leakage reduction techniques in sub-threshold FPGA architectures in an effort to guide development of such architectures toward optimal power savings. Specifically, we will use an existing sub-threshold FPGA architecture designed by Joseph Ryan and Ben Calhoun as our implementation platform. We will implement and test variations on 4 known

leakage reduction techniques in various parts of the FPGA and compare their power savings, area overhead, and delay overhead.

The leakage reduction techniques to be tested are outlined below:

Stacking

The stacking method allows a designer to bias the source of a transistor by placing multiple “off” devices in series with that transistor. This method has 3 major effects that lead to a reduction of leakage current: first, an increased source voltage decreases Vds and in turn decreases leakage current due to DIBL effect; second, because of the body effect, source voltage increases, leading to an increase in threshold voltage and and an exponential decrease in leakage current; finally, this method leads to a lower Vgs which, in turn, leads to an additional exponential decrease in leakage current. The body effect and effect of lowered Vgs are primary contributors to the effectiveness of this method. There is some overhead with this method however, and it can be difficult to determine the optimal placement of stacked structures [6].

Dual Vt Partitioning

Leakage currents have an exponential dependence on device threshold voltage. [6] shows that reducing a device's threshold voltage by about 85 mV increases the leakage current through that device by an order of magnitude. Though high threshold voltages are preferred for leakage reduction, a transistor with high threshold voltage is slower than one with low threshold voltage. Partitioning a circuit into blocks of low threshold devices and blocks of high threshold devices can provide the advantages of both. By using low threshold transistors in critical paths, a designer can maintain desired switching speeds. By using high threshold devices in non-critical paths, a designer can also reduce leakage. This approach can provide an optimal median point between leakage reduction and switching speed, and has been shown to be effective in super-threshold FPGAs [5].

Multi-threshold CMOS (MTCMOS)

MTCMOS is very similar to dual-Vt partitioning, but uses combinations of high and low threshold devices in every element rather that partitioning those elements. Essentially, high threshold “sleep transistors” are placed between the rails and the desired low threshold logic block. When the block is in standby, or asleep, the sleep transistors are off and the total leakage current through the block is reduced to the leakage current of the sleep transistors. Since the sleep transistors have a high threshold, this current is very small. When the block returns to work, the sleep transistors are turned on and, if sized properly, provide rail voltages to the interior block. This method has been found to be very effective at reducing standby leakage currents [5]. Note that this method also allows designers to maintain the speed of the low threshold logic blocks while achieving the leakage power savings that come with high threshold devices. This method can be complex to implement; [8] presents some methods for determining optimal MTCMOS implementation. Additionally, this method requires special consideration when dealing with sequential circuits; [9] presents recommendations for

implementing MTCMOS methods in sequential circuits that we will include in our implementations.

Variable threshold CMOS

Traditional CMOS implementations connect the body terminal of transistors to the appropriate rail in order to avoid the effects of body biasing. However, body biasing can provide an effective method for adjusting threshold voltage. This allows designers to both reduce leakage current in standby mode and improve performance in active mode by allowing them to dynamically change the threshold voltage of the device [6]. In most cases, such a variable threshold CMOS device is implemented using a triple well process. However, in sub-threshold operating regimes, one may be able to dynamically connect the body to the appropriate rails. In super-threshold operation, this might destroy the device; in sub-threshold operation, rail voltages are low enough that the device will not be destroyed.

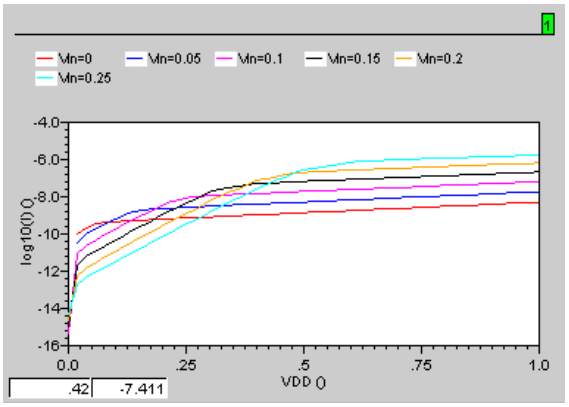

Figure 1. Current through a simple inverter at various input voltages. Plot is log10(Ip) vs VDD.

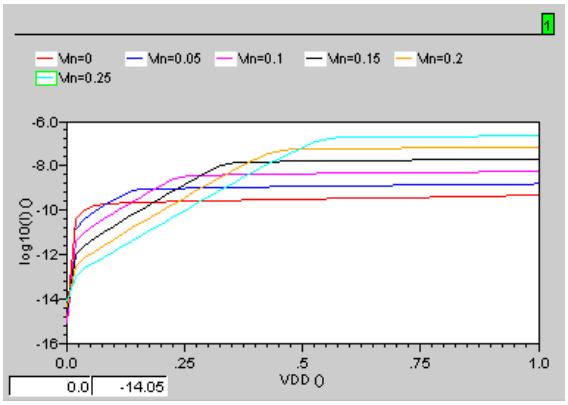

Figure 2. The same inverter with stacking. Note the differing scales for current; stacking decreases sub-threshold leakage current significantly.

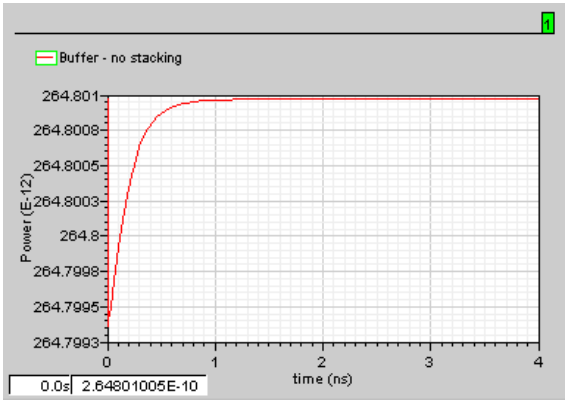

Figure 3. Power dissipation through a simple, inactive buffer. After a certain initial setup, the power becomes constant at it's leakage value.

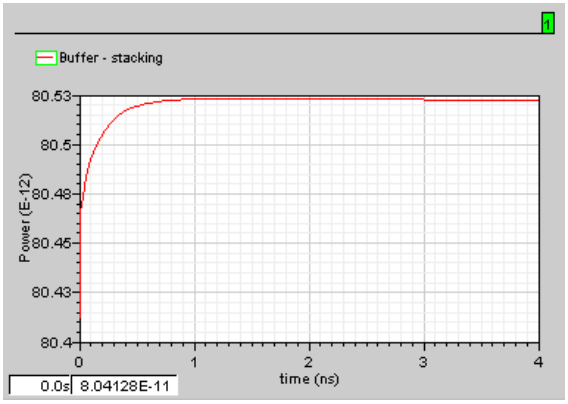

Figure 4. The same buffer with stacking. Note the differing time scales; stacking significantly decreases leakage power consumption.

Expected Outcomes

Upon completing this project, we hope to identify and implement variations of the techniques enumerated above that will provide the maximum standby leakage current reduction for sub-threshold FPGAs. We expect to find that variations of the techniques enumerated above will, in fact, effectively reduce standby leakage current in the chosen FPGA architecture. Much work of this type has been performed for super-threshold FPGAs, we hope to contribute the same quality of work for sub-threshold FPGAs to the greater research community. Finally, we hope to propose and test novel variations of these techniques specific to sub-threshold operation that have not been previously investigated in super-threshold regimes.

Project Timeline and Task Breakdown

We plan to follow the attached timeline in completing this project. Note that all of our implementation and simulation will be done in the Ocean simulation tool. Also note that all simulations after the initial proposal date will be included in the final report.

Week Beginning	Task	Investigator
10/05/09	- Investigate sub-threshold operation in general circuits - Become familiar with Ocean as a simulation tool - Meet with Joe Ryan, Professor Calhoun to discuss FPGA architecture **- Prepare for design review 1**	Runjie, Mike
10/12/09	- Build and simulate basic implementations of leakage reduction methods, 1 each **- Prepare project proposal**	Runjie, Mike
10/19/09	- Convert CLB architectures to use available PTMs, 1 each (both CLB_cstm, CLB_base2) - Simulate functionality, leakage characteristics of unmodified CLBs, CLB components	Runjie, Mike
10/26/09	- Implement initial designs for MTCCMOS and stacking leakage reduction methods - Run intitial simulations CLBS with leakage reduction methods in place	Runjie
	- Implement initial designs for dual-Vt CMOS leakage reduction methods - Run intitial simulations CLBS with leakage reduction methods in place	Mike
11/02/09	- Begin optimizing respective designs for leakage reduction methods - Meet with Joe Ryan, Professor Calhoun to discuss initial results and determine further procedures **- Prepare Design Review 2**	Runjie, Mike
11/09/09	- Finalize CLB-specific leakage reduction implementations based on feedback from Joe Ryan and Professor Calhoun; simulate final leakage reduction implementations - Begin investigating interconnect leakage	Runjie, Mike
11/16/09	- Implement interconnect leakage reduction techniques - Meet with Joe Ryan, Professor Calhoun to discuss interconnect leakage reduction techniques	Runjie, Mike
11/23/09	- Finalize interconnect leakage reduction implementations based on received feedback; simulate final interconnect leakage implementations, full FPGA leakage characteristics	Runjie, Mike
11/30/09	**- Prepare Final Deliverables**	Runjie, Mike

Additional Simulations

In an effort to familiarize ourselves with sub-threshold operation in general, we also assembled the following simulations:

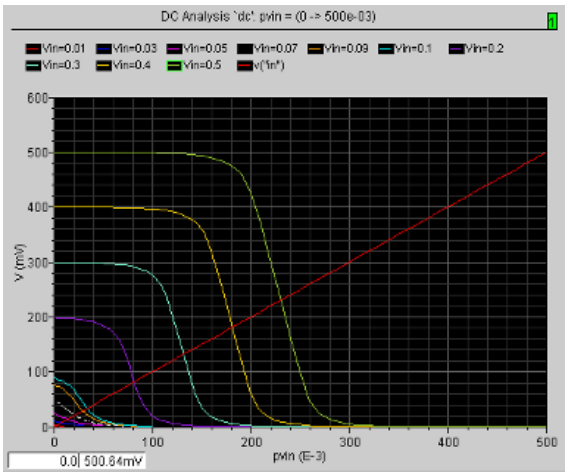

Figure 5. The VTC of an inverter operating in the sub-threshold region. Note that the gain noise margins are prominent until ~200mV.

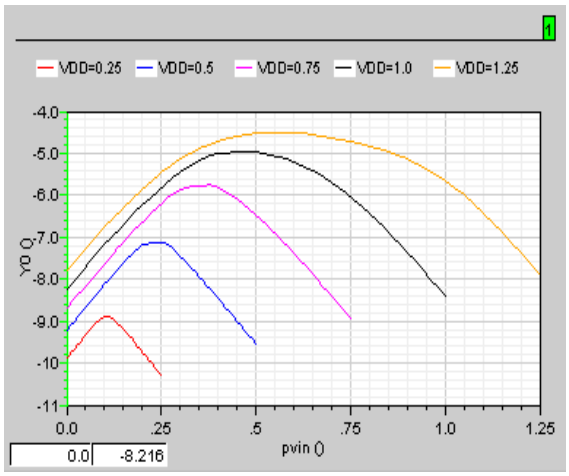

Figure 6. Power dissipation vs. Vin for various values of VDD. Note that power dissipation, both dynamic and leakage, decreases as driving voltage decreases.

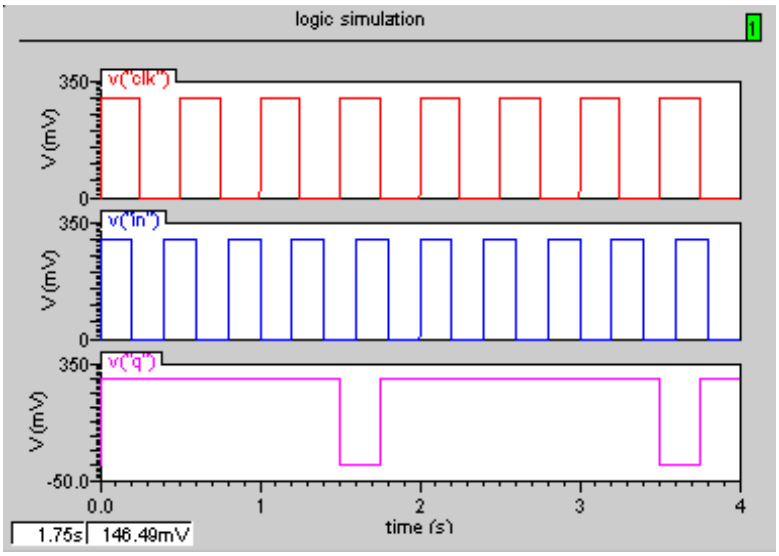

Figure 7. A simple D flip-flop operating at sub-threshold voltages (300mV), albeit very slowly. Note the time scale (s).

References

[1] B. Calhoun, A. Wang and A. Chandrakasan, “Modeling and Sizing for Minimum Energy Operation in Subthreshold Circuits”, IEEE Journal of Solid-State Circuits, Vol. 40, No.9, 2005.
[2] T. Tuan and B. Lai, “Leakage Power Analysis of a 90nm FPGA”, IEEE Custom Integrated Circuits Conf, pp.57-60, 2003.
[3] F. Li, D. Chen, L. He and J. Cong, “Architecture Evaluation for Power-Efficient FPGAs”, FPGA, pp 175-184, 2003.
[4] K. Poon, A. Yan and S. Wilton, “A Flexible Power Model for FPGAs”, FPL 2002, LNCS 2438, pp. 312-321, 2002
[5] A. Kumar and M. Anis, “Dual-Vt Design of FPGAs for Subthreshold Leakage Tolerance”, ISQED, pp 735-740, 2006.
[6] J. Kao, S. Narendra and A. Chandrakasan, “Subthreshold Leakage Modeling and Reduction Techniques”, ICCAD, pp.141-148, 2002.
[7] B.H. Calhoun, S. Khanna, R. Mann, J. Wang, "Sub-threshold circuit design with shrinking CMOS devices," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on , vol., no., pp.2541-2544, 24-27 May 2009.
[8] B.H. Calhoun; F.A. Honore; A.P. Chandrakasan, "A leakage reduction methodology for distributed MTCMOS," Solid-State Circuits, IEEE Journal of , vol.39, no.5, pp. 818-826, May 2004
[9] J. Kao; A. Chandrakasan, "MTCMOS sequential circuits," Solid-State Circuits Conference, 2001. ESSCIRC 2001. Proceedings of the 27th European , vol., no., pp. 317-320, 18-20 Sept. 2001